U.S. AIR FORCE

Team 6: Continuous ATO

Integrity - Service - Excellence

U.S. AIR FORCE

Continuous ATO

- Team Lead: Rob Vietmeyer (DoD CIO) and Nick Chaillan – SAF/CSO and DSAWG DevSecOps chair.
- Deliverables:
  - **(MVP: PRIORITY):** Continuous ATO guidance:
    - Section 1: How to authorize the Platform's **PROCESS** (Continuous Integration/Continuous Delivery (Software Factory)) **with mandated testing and security gates.** The software coming out of the factory and that is RUNNING IN PRODUCTION **on the Platform** (Kubernetes with SCSS) also benefits from the cATO.
      - **We authorize layers so they can be swappable and environments can be dynamic:**
        - Infrastructure, Platform (K8S + SCSS + EFK), Service Mesh, App/Microservices + enabling layers with Continuous Monitoring, Hardened Containers, CI/CD (with GitOps) and Enterprise Services.
    - Section 2: How to certify **TEAMS** using the Platform so they can produce quality software and be trained to move to DevSecOps
    - Appendix 1: list he expected deliverables / artifacts of pipelines/platforms
    - Appendix 2: How to automate artifacts push into eMass. Is eMass still the way to go?

U.S. AIR FORCE

What is a Continuous ATO?

- A Continuous ATO is very different from a traditional ATO or a Fast-Track/Accelerated ATO:
  - Platforms have to be compliant with the DoD Enterprise DevSecOps Ref Design to ensure DoD-wide reciprocity, including the use of the Sidecar Container Security Stack (SCSS). Platform controls are mapped to NIST-800-53.
  - We authorize the Platform's **PROCESS** (Continuous Integration/Continuous Delivery (Software Factory)) **with mandated testing and security gates.** The software coming out of the factory and that is RUNNING IN PRODUCTION **on the Platform** (Kubernetes with SCSS) also benefits from the cATO.
  - **We authorize layers so they can be swappable and environments can be dynamic:**
    - Infrastructure, Platform (K8S + SCSS + EFK), Service Mesh, App/Microservices + enabling layers with Continuous Monitoring, Hardened Containers, CI/CD (with GitOps) and Enterprise Services.
  - We authorize **TEAMS** using the Platform so they can produce quality software and be trained to move to DevSecOps
  - A key principle of DevSecOps is the **baked-in security** with:
    - Zero Trust
    - Automation
    - Removal of environment drifts
    - Behavior Detection
    - Continuous Monitoring
    - Pen-testing